



UNIVERSITAS BUNDA MULIA  
FAKULTAS TEKNOLOGI DAN DESAIN  
ISSN 1979-1496

## JURNAL TEKNOLOGI INFORMASI

Vol. 10 No. 1 Juni 2014

Aplikasi Untuk Pemantauan LAN Pada Studi Kasus Di Universitas Bunda  
Mulia  
Halim Agung

Analisa Dan Perancangan Sistem Informasi Rekam Medik Pada Klinik drg.  
Lily Sunaryo  
Giovani Henry, I Gusti Ngurah Suryantara

Fitur Matriks Populasi Piksel Untuk Membedakan Frame-frame Dalam Deteksi  
Gerakan  
Teady Matius Surya Mulyana

Penerapan Algoritma Teorema Bayes Pada Aplikasi Pengejaan Kata  
Marta Lenah Haryanti, Dwi Fegianata

Penerapan Metode Greedy Coloring Dalam Menyelesaikan Masalah  
Persimpangan Jalan  
Faisal

Steganografi Dengan Deret Untuk Mengacak Pola Penempatan Pada RGB  
Renddy, Teady Matius Surya Mulyana

Nama Jurnal | Vol. 10 No. 1 | Juni 2014 | ISSN 1979-1496



UNIVERSITAS BUNDA MULIA  
FAKULTAS TEKNOLOGI DAN DESAIN

Jl. Lodan Raya No. 2, Ancol Jakarta Utara 14430  
Tlp. 021-6909090 | Fax. 021-6909712  
www.ubm.ac.id



ISSN 1979-1496

9 771979 149663



Teknologi Informasi | Vol. 10 No. 1 | Juni 2014

**JURNAL TEKNOLOGI INFORMASI  
JURUSAN TEKNIK INFORMATIKA, UNIVERSITAS BUNDA MULIA  
Volume 10, Nomor 1, Juni 2014**

**PENGELOLA JURNAL**

**PELINDUNG  
Rektor  
Universitas Bunda Mulia**

**PENANGGUNG JAWAB  
Wakil Rektor  
Bidang Akademik**

**PENYUNTING UTAMA  
Bambang Dwi Wijanarko**

**PENYUNTING AHLI  
Lukman Hakim  
Teady Matius Surya Mulyana  
Henny Hartono**

**SEKRETARIAT  
Henny Hartono**

**ALAMAT  
Jurusan Teknik Informatika  
Universitas Bundamulia  
Jl.Lodan Raya No.2  
Jakarta Utara 14430  
Telpon (021) 6909090  
Fax (021) 6909712**

**EMAIL  
[jurnal-tekinfo@bundamulia.ac.id](mailto:jurnal-tekinfo@bundamulia.ac.id)**

**WEBSITE  
[WWW.BUNDAMULIA.AC.ID](http://WWW.BUNDAMULIA.AC.ID)**

## DAFTAR ISI

APLIKASI UNTUK PEMANTAUAN LAN PADA STUDI KASUS DI UNIVERSITAS BUNDA MULIA Halim Agung	1
ANALISA DAN PERANCANGAN SISTEM INFORMASI REKAM MEDIK PADA KLINIK DRG. LILY SUNARYO Giovani Henry, I Gusti Ngurah Suryantara	8
FITUR MATRIKS POPULASI PIKSEL UNTUK MEMBEDAKAN FRAME-FRAME DALAM DETEKSI GERAKAN Teady Matius Surya Mulyana	13
PENERAPAN ALGORITMA TEOREMA BAYES PADA APLIKASI PENGEJAAN KATA Marta Lenah Haryanti, Dwi Fegiannata	19
PENERAPAN METODE GREEDY COLORING DALAM MENYELESAIKAN MASALAH PERSIMPANGAN JALAN Faisal	22
STEGANOGRAFI DENGAN DERET UNTUK MENGACAK POLA PENEMPATAN PADA RGB Renddy, Teady Matius Surya Mulyana	28

**JURNAL TEKNOLOGI INFORMASI**  
**JURUSAN TEKNIK INFORMATIKA, UNIVERSITAS BUNDA MULIA**  
**Volume 10, Nomor 1, Juni 2014**

**PENGANTAR REDAKSI**

Puji dan Syukur dipanjatkan kepada Tuhan Yang Maha Esa atas karunia dan Lindungan-Nya sehingga Jurnal ***Teknologi Informasi*** terbitan tahun ke lima ini bisa terwujud. Diharapkan dengan jurnal ini dapat membantu menyebarkan hasil-hasil penelitian yang terkait dengan permasalahan teknologi informasi. Pada kesempatan ini Tim Redaksi sangat mengharapkan partisipasi para dosen dan tenaga peneliti untuk sudi menyumbangkan naskah guna diterbitkan pada jurnal ***Teknologi Informasi***, dan terima kasih atas partisipasi, perhatian serta kerjasamanya dari berbagai pihak sehingga dapat menerbitkan Jurnal ***Teknologi Informasi*** ini.

Jakarta, Juni 2014

Dewan Penyunting

## Fitur Matriks Populasi Piksel Untuk Membedakan *Frame-frame* Dalam Deteksi Gerakan

Teady Matius Surya Mulyana  
[tmulyana@bundamulia.ac.id](mailto:tmulyana@bundamulia.ac.id), [teadymatius@yahoo.com](mailto:teadymatius@yahoo.com)  
Teknik Informatika Universitas Bunda Mulia

### *Abstrak*

Pengenalan gerakan tangan secara optis harus mengidentifikasi apakah suatu gerakan sedang dalam fase bergerak atau sudah mencapai status yang sudah *fix*. Pada proses ini diperlukan suatu fitur yang mampu menjadi penciri sebuah *frame* dengan *frame* lainnya

Matriks Populasi Piksel merupakan sekumpulan fitur dalam bentuk sel-sel matrik yang masing-masing berisi populasi piksel hitam berbanding jumlah piksel sel matriks populasi piksel pada citra. Fitur-fitur ini nantinya akan dihitung jarak antara sebuah *frame* dengan *frame* lainnya.

Sebagai bagian dari *computer vision*, gerakan tangan ini dapat diproses dengan menggunakan pengukuran jarak dengan *L1-metric* untuk mendapatkan perbedaan antara *frame* yang satu dengan *frame* lainnya berdasarkan *threshold* yang diberikan.

Proses ini hanya akan menemukan perbedaan antara akan memerlukan bantuan teknik lain yaitu penggunaan *stack*. Tetapi untuk membedakan *frame* yang satu dengan *frame* yang lain sudah cukup memadai.

**Kata Kunci:** *matriks populasi piksel, L1-metric, membedakan frame*

### PENDAHULUAN

Low (Low, 1991), dalam bukunya yang berjudul *Introductory Computer Vision and Image Processing* membagi pixel-pixel penyusun citra kedalam sel-sel matriks. Persentase pixel-pixel terhadap sel-sel matriks itu yang menjadi *pattern vector* yang akan dipergunakan untuk mengenali suatu obyek citra. Low juga menggunakan sel matriks 4 x 4 untuk mengenali huruf kanji[2].

Matriks populasi piksel adalah sekumpulan sel yang terbentuk dari pembagian citra-citra membentuk kolom dan baris. Masing-masing sel berisi populasi piksel pada bagian-bagian citra yang diwakilinya. Matriks populasi piksel dapat tersusun dari

matriks 2x2, 3x3, 4x4, 5x5, 6x6 dan seterusnya.

Mulyana (Mulyana, 2006), dalam laporan tesisnya menjelaskan bahwa untuk dapat mengakomodasi penyebaran piksel obyek citra sebaiknya menggunakan matriks yang dapat mengakomodasi penyebaran piksel-piksel citra obyek tersebut[4].

Lu (Lu, 1999), dalam bukunya yang berjudul *Multimedia Database Manajemen System* menjelaskan untuk mengenai pengukuran jarak antara dua buah citra dapat dipergunakan dengan menggunakan rumus *L1-Metric*[3].

*L1-metric* melakukan pengukuran jarak antara fitur-fitur yang dimiliki dua buah citra. Dimana jarak kedua buah citra ini yang nantinya akan

dipertimbangkan sebagai kemiripan antara dua buah citra. Semakin kecil nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin mirip. semakin besar nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin berbeda.

Rumus dari *L1-Metric* diperagakan pada rumus (1).

$$d(I, H) = \sum_{l=1}^n |i_l - h_l| \dots\dots\dots (1)$$

Notasi-notasi yang dipergunakan pada rumus (1) adalah :

- $l$                       Pencacah fitur
- $n$                       Jumlah fitur
- $I$                       Himpunan fitur citra pada top *stack* / citra yang terakhir disimpan
- $i$                       Fitur citra pada top *stack* / citra yang terakhir disimpan
- $H$                       Himpunan fitur citra yang akan diuji
- $h$                       Fitur citra yang akan diuji
- $D(I, H)$               Jarak citra  $I$  terhadap citra  $H$

## METODE

Metode untuk melakukan pengenalan gerakan tangan secara optis ini terdiri dari proses akuisisi citra, *pre-processing*, ekstrasi fitur dan pengenalan citra. Setiap proses akan berkesinambungan antara satu proses dengan proses lainnya.

## Akuisisi Citra

Akuisisi citra didapat dari kamera web, dimana citra yang ditangkap kamera akan *dicapture* setiap *frame* tertentu. Hal ini

dimaksudkan agar tidak semua *frame* yang tertangkap kamera diproses. Alasan dari proses ini adalah karena untuk setiap beberapa *frame* hanya akan memiliki kesamaan posisi yang relatif. *Frame* yang berhasil ditangkap akan disimpan bitmapnya pada array.

## Pre-Processing

Pre-processing dimaksudkan untuk mendapatkan citra yang dianggap layak dianalisa oleh komputer. Karena penganalisa citra adalah komputer, maka citra yang memungkinkan untuk diproses harus berupa citra biner. Sutoyo (Sutoyo, 2009) menjelaskan citra dibinerisasi dengan beberapa cara. Cara yang paling mudah adalah dengan menjadikan citra grayscale dan menentukan batas hitam dan putih, sehingga nilai grayscale dibawah batas putih akan dijadikan 0, dan nilai grayscale yang sesuai dengan batas putih akan dijadikan nilai 1 atau nilai grayscale 255[7].

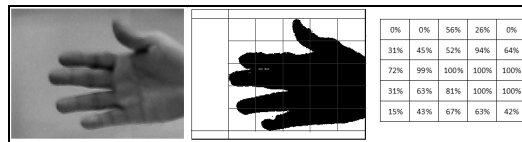
## Ekstrasi Fitur

Setelah didapatkan citra biner dari *frame* yang akan diproses, berikutnya dilakukan ekstrasi fitur. Fitur yang dipergunakan pada percobaan ini adalah fitur populasi matriks 5 x 5. Dimana citra dibagi pada sel-sel dalam suatu matriks dua dimensi berukuran 5 x 5. Setiap sel akan berisi nilai yang merupakan populasi piksel dari setiap area citra yang merupakan luas dari 1/25 luas citra dalam hitungan piksel.

Proses ekstraksi fitur terdiri dari proses pencarian batas atas, batas kiri, batas kanan dan batas bawah. Mulyana (Mulyana, 2006) menjelaskan proses pencarian batas atas, batas bawah, batas kanan dan batas bawah dapat dilakukan dengan scanline secara horisontal dan secara vertikal. Untuk mendapatkan batas kiri objek dapat dilakukan scanline horisontal kolom perkolom mulai dari kiri citra sampai ditemukan kolom pertama yang berisi

piksel hitam. Untuk mendapatkan batas kanan objek dapat dilakukan scanline horisontal kolom perkolom mulai dari kanan citra sampai ditemukan kolom pertama yang berisi piksel hitam[5].

Untuk mendapatkan batas atas objek citra dilakukan scanline vertikal baris per baris, mulai dari atas citra sampai ditemukan baris pertama yang berisi piksel hitam. Untuk mendapatkan batas bawah objek citra dilakukan scanline vertikal baris per baris, mulai dari bawah citra sampai ditemukan baris pertama yang berisi piksel hitam.



Gambar 1. Contoh Ekstrasi Fitur

Gambar 1 merupakan contoh ekstrasi fitur. Gambar pertama adalah citra berwarna, gambar di tengah merupakan citra biner. Serta gambar paling kanan merupakan populasi piksel hasil ekstrasi fitur pada masing-masing sel nya.

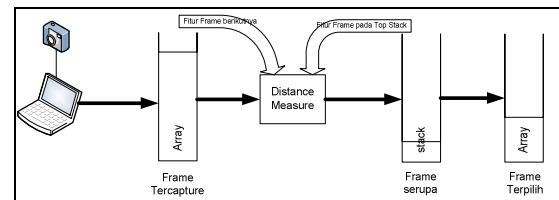
Pada gambar yang di tengah garis yang berwarna hitam gelap merupakan batas kiri, batas atas dan batas bawah objek citra, sedangkan batas kanan menyatu dengan tepi kanan citra.

Garis yang berwarna abu-abu merupakan batas-batas fitur matriks populasi piksel. Terdapat 5 baris sel yang masing-masing baris berisi 5 sel, sehingga membentuk matriks 5 x 5.

Pada proses pengenalan citra, setiap *frame* citra yang *tercapture* akan diukur jaraknya dengan citra sebelumnya yang masuk ke dalam penampung citra yang diuji sampai dengan memenuhi persyaratan yang sudah ditentukan. Jika distance dengan *frame* *tercapture*  $\leq$  *treshold* maka akan dianggap sama dengan yang sudah tersimpan.

Mulyana (Mulyana, 2013) dalam laporan penelitiannya menjelaskan proses ini akan dibedakan tiga kelompok *frame* citra[6], kelompok pertama adalah *frame* yang *tercapture*, yaitu *frame* yang *dicapture* dari kamera dengan aturan setiap sejumlah *frame* yang tertangkap kamera, maka citra pada *frame* tersebut diambil dan disimpan pada sebuah array. Citra akan disimpan bitmap serta fitur-fitur populasi matriks nya.

Kelompok kedua adalah *frame* yang masuk ke dalam *stack*. Pada *stack* akan disimpan nomor indeks *frame* yang mengalami *frame* citra tersebut pada array *frame* citra *tercapture*. Pada percobaan ini setiap kali *stack* mencapai 6 jumlah enam *frame* citra, maka citra tersebut akan disimpan pada kelompok *frame* citra terpilih. Ilustrasi dari urutan kelompok *frame* tersebut dapat dilihat pada gambar 2.



Gambar 2. Proses Pemilihan Frame

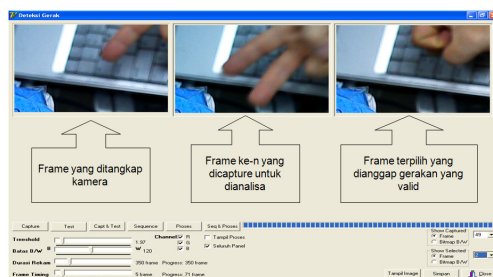
Pada gambar 2 diilustrasikan *frame* *tercapture* tersimpan pada array dengan jumlah tumpukan lebih banyak daripada lainnya, dengan asumsi *frame* *tercapture* *n* *frame* yang ditentukan akan memiliki *frame* lebih banyak daripada *stack* *frame* serupa dan *frame* terpilih. Sedangkan *stack* *frame* serupa memiliki tumpukan yang terbatas, karena hanya akan dipergunakan sebagai tempat penampungan sampai mencapai sejumlah *frame* tertentu yang dianggap mirip, kemudian setelah jumlah yang sudah ditentukan tersebut tercapai, maka *frame* yang pertama masuk ke dalam *stack* akan dimasukkan ke dalam array *frame* terpilih. Tidak semua *frame* pada *frame* *tercapture* akan dipilih ke dalam *frame* terpilih.

Antara *frame* *tercapture* dengan *stack frame* serupa terdapat proses distance measure (pengukuran jarak) menggunakan rumus *L1-metric*. Untuk menghemat memori dan mempercepat pemrosesan, maka pada *Array frame* *tercapture* akan disimpan bitmap beserta fitur matriks populasi pikselnya, sedangkan pada *stack frame* serupa dan *array frame* terpilih, akan disimpan alamat indeks *frame* pada *array frame* *tercapture*.

### PEMBAHASAN

Sebagai gambaran proses dapat dilihat pada contoh tampilan perangkat lunak pada gambar 3. Interface aplikasi tersebut menampilkan *frame-frame* citra yang diproses, disediakan tiga panel tampilan *frame*:

- Bagian tampilan *frame* yang ditangkap kamera ada pada panel paling kiri
- Bagian tampilan *frame* ke *n* yang *dicapture* untuk dianalisa terdapat pada panel di tengah
- Bagian tampilan *frame* terpilih yang dianggap gerakan valid terdapat pada panel paling kanan.



Gambar 3. Tampilan Jendela Program

Pengujian teknik ini dilakukan dengan menggunakan perangkat lunak yang sudah dibuat yang menerapkan metode penggunaan *stack* dan matriks populasi piksel. Dimana pengujian dilakukan dengan mengkombinasikan beberapa gerakan

tangan dengan tresshoold yang berbeda.

Ada dua macam citra yang diuji pada pengujian ini, yaitu citra yang *backgroundnya* dihilangkan, sehingga matriks populasi piksel hanya dihitung pada objek dan citra dengan *background* yang ikut dibinerkan.

Untuk menghilangkan *background* dilakukan dengan kombinasi penggabungan *cannel R, G* dan *B* dan pengaturan batas hitam dan putih, dengan asumsi dengan kondisi pencahayaan tertentu akan ada *cannel* tertentu yang menjadi dominan mempengaruhi warna objek sehingga *channel* tersebut perlu dihilangkan. Untuk citra dengan *background*, matriks populasi piksel di hitung mulai dari tepi-tepi citra.

Gambar 4 menunjukkan contoh citra-citra yang ditangkap oleh kamera serta sudah dibinarisasi menjadi citra biner. Sedangkan gambar 5 menunjukkan contoh citra yang terpilih sebagai citra dengan gerakan yang *fix*.

Dari gambar 4 terdapat *frame* *capt1.jpg* sampai *capt16.jpg* yang merupakan *frame* yang memiliki kemiripan yang tidak mencapai enam *frame*. Sehingga dianggap *frame* peralihan. Sampai akhirnya didapat *frame* *capt17.jpg*.

*Frame* *select1.jpg* adalah *frame* yang didapat dari *frame* *capt17.jpg*, dan merupakan hasil pengukuran dari *capt17.jpg* sampai *capt23.jpg* yang mempunyai tujuh *frame* yang dianggap mirip. *Frame* *capt24.jpg* sampai *capt28.jpg* merupakan *frame* peralihan sampai didapat *frame* *capt29.jpg*.

*Frame* *select2.jpg* merupakan *frame* yang didapat dari *capt29.jpg* dan merupakan hasil pengukuran dari *capt29.jpg* s/d *capt36.jpg* yang mempunyai tujuh *frame* yang dianggap mirip. *Frame* *capt37.jpg* merupakan *frame* peralihan dari *capt36.jpg* ke *capt38.jpg*, dimana *capt38.jpg* merupakan *frame* yang dipilih yang menjadi *frame* *select3.jpg*, dimana *select3.jpg* merupakan *frame* yang

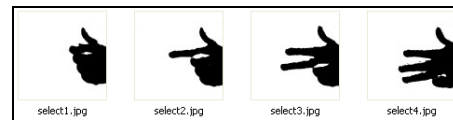


didapat dari capt38.jpg dan merupakan hasil pengukuran dari capt38.jpg s/d capt48.jpg yang mempunyai sebelas *frame* yang dianggap mirip. *Frame* capt49.jpg sampai *frame* capt51.jpg merupakan *frame* peralihan sampai didapat *frame* capt52.jpg yang merupakan *frame* terpilih menjadi select4.jpg yang didapat dari hasil pengukuran *frame* capt52.jpg sampai *frame* capt62.jpg yang mempunyai kemiripan *frame* yang satu dengan *frame* berikutnya sebanyak sebelas *frame*. Sedangkan sisanya *frame* capt63.jpg sampai *frame* capt70.jpg tidak memiliki kesamaan sampai enam *frame*.

*Stack* akan dipergunakan untuk membantu dalam proses penghitungan jumlah *frame* yang mempunyai kemiripan satu dengan lainnya. *Stack* dimanfaatkan sebagai penyimpanan obyek berupa serangkaian fitur matriks populasi piksel. Dengan sifat-sifat yang ada pada *stack* dan operasinya, maka *stack* dan operasinya dengan rule-rule yang disusun akan dimanfaatkan untuk menampung fitur-fitur matriks populasi piksel serta dihitung jaraknya dengan *frame-frame* berikutnya sehingga didapatkan *frame* yang dianggap sama dengan *frame* berikutnya, serta mengabaikan *frame* yang berbeda dengan *frame* berikutnya.



Gambar 4. Citra Hasil *Capture*



Gambar 5. Citra yang terpilih

## Hasil Pengujian

Hasil pengujian terdapat pada lampiran. Contoh data yang dihasilkan dan dipakai pada pengujian adalah seperti yang diperagakan pada tabel 1 dan tabel 2. Pada tabel-tabel ini tercantum jumlah gerakan yang diasumsikan oleh penguji kemudian jumlah gerakan yang dikenali dengan *treshold* tertentu. Dari hasil pengujian tanpa *background* dengan *treshold* yang berbeda, menghasilkan ketepatan pengenalan sebanyak 84% dimana ketidak tepatan disumbangkan oleh *treshold* 1.97 dan 4.85. *Treshold* 4.85 menghasilkan ketepatan hanya 20%.

Tabel 1. Hasil Pengujian Tanpa *Background*

Percobaan	Jum Gerakan	Treshold	dikenali	persentase
1	5	1.97	4	80%
2	5	1.97	5	100%
3	3	1.97	3	100%
4	5	0.66	5	100%
5	4	1.1	4	100%
6	5	1.97	3	60%
7	5	4.82	1	20%
8	4	0.66	4	100%
9	4	1.97	4	100%
Rata2				84%

Tabel 2 merupakan hasil pengujian dengan *background*. Dari hasil pengujian, didapat bahwa pada *treshold* tertentu didapatkan ada pengenalan gerakan yang akurat mencapai 100%, itu terjadi pada *treshold* 0.66 sampai 3.51, pada *treshold* 1.97 dan 2.85 memiliki rata-rata keberhasilan yang besar, sedangkan pada *treshold* yang lebih kecil atau lebih besar memiliki keberhasilan yang lebih kecil.

**Tabel 2. Hasil Pengujian Dengan Background**

Perco- baan	Jum Gerakan	Jum Gerakan yg dikenali dg Treshold						% Keberhasilan pada masing2 treshold					
		0.66	1.10	1.97	2.85	3.51	4.82	0.66	1.10	1.97	2.85	3.51	4.82
1	6	2	3	5	3	3	6	33%	50%	83%	50%	50%	100%
2	7	3	4	4	4	4	2	43%	57%	57%	57%	57%	29%
3	7	1	3	5	5	4	2	14%	43%	71%	71%	57%	29%
4	4	4	5	4	3	2	2	100%	125%	100%	75%	50%	50%
5	6	1	2	4	4	5	4	17%	33%	67%	67%	83%	67%
6	8	3	4	6	5	4	4	38%	50%	75%	63%	50%	50%
7	7	1	2	5	5	5	4	14%	29%	71%	71%	71%	57%
8	5	0	3	4	6	5	3	0%	60%	80%	120%	100%	60%
9	8	2	5	3	2	2	1	25%	63%	38%	25%	25%	13%
10	7	6	7	7	3	2	2	86%	100%	100%	43%	29%	29%
Rata 2								37%	61%	74%	64%	57%	48%

Melihat tersebarnya angka keberhasilan 100% yang tersebar pada semua *treshold*, maka perlu dilakukan pengujian dengan percobaan yang lebih banyak dengan variasi *treshold* yang lebih bervariasi, sehingga didapatkan kondisi yang benar-benar mencerminkan keakuratan metode ini.

Pengujian tanpa *background* menghasilkan keberhasilan 100% yang lebih banyak daripada pengujian dengan *background*. Hal ini disebabkan karena perubahan dari objek yang seharusnya di kenali hanya menempati 9 sampai 12 sel dari matriks populasi piksel, sedangkan sisanya ditempati oleh populasi piksel dari *background* yang tidak akan berubah. Dengan demikian mengakibatkan kurang akuratnya keakuratan hasil pengujian.

## SIMPULAN

Kesimpulan dari pemanfaatan *stack* untuk mengenali adanya gerakan tangan:

- Matriks populasi piksel dengan menggunakan *L1-Metric* mampu mengenali adanya perbedaan antara *frame-frame*.

## Saran

Saran untuk melengkapi kelanjutan metode ini:

- Diperlukan penelitian lebih lanjut untuk memisahkan obyek tangan dengan latar belakang.

## DAFTAR PUSTAKA

- [1] J, Simarmata dan T. Chandra, "Grafika Komputer", Andi Offset, Yogyakarta, 2007.
- [2] Low, Adrian., *Introductory Computer Vision and Image Processing*, McGraw-Hill, Berkshire, UK, 1991.
- [3] Lu, Guojun., *Multimedia Database Management Systems*, Artech House, London, 1999.
- [4] Mulyana, Teady Matius Surya., *Penggunaan Matriks Populasi Pixel Dan Relational Database Untuk Mengenali Huruf Kanji Dan Radikalnya*, Masters Thesis, Computer Science Study Program, Gadjah Mada University, 2006.
- [5] Mulyana, Teady Matius Surya., & Harjoko, Agus., "A Chinese Character Recognition Method Based On Population Matrix and Relational Database", *Proceeding of Information & Communication Technology Seminar, IEEE (P518-523)*, 2006.
- [6] Mulyana, Teady Matius Surya., *Pengenalan Gerakan Tangan Secara Optis*, Laporan Penelitian, Universitas Bunda Mulia, Jakarta, 2013.
- [7] Sutoyo, T., mulyanto, E., Suhatono V., Nurhayanti OD., Wijanarto., *Teori Pengolahan Digital*, Andi Offset, Yogyakarta, 2009.